

王道计算机考研  
www.cskaooyan.com

本节内容

静态链表

王道考研/CSKAOYAN.COM

1

知识总览

静态链表

什么是静态链表

如何定义一个静态链表

简述基本操作的实现

王道考研/CSKAOYAN.COM

2



### 什么是静态链表

**静态链表**

addr	数据元素	游标 (下一个结点的数组下标)
0	头	2
1	e <sub>2</sub>	6
2	e <sub>1</sub>	1
3	e <sub>4</sub>	-1
4		
5		
6	e <sub>3</sub>	3
7		
8		
9		

内存

单链表：各个结点在内存中星罗棋布、散落天涯。

静态链表：分配一整片连续的内存空间，各个结点集中安置。

每个数据元素 4B，每个游标4B（每个结点共 8B）  
设起始地址为 **addr**

e<sub>1</sub> 的存放地址为 **addr + 8\*2**

单链表

```

L → [头] → [e1] → [e2] → [e3] → [e4] → NULL
            
```

数据元素      指向下一个结点的指针（地址）

王道考研/CSKAOYAN.COM

3

### 用代码定义一个静态链表

**a**

0
1
2
3
4
5
6
7
8
9

内存

```

#define MaxSize 10
struct Node{
    ElemType data;
    int next;
};

void testSLinkList() {
    struct Node a[MaxSize];
    //.....后续代码
}
    
```

//静态链表的最大长度  
//静态链表结构类型的定义  
//存储数据元素  
//下一个元素的数组下标

数组 a 作为静态链表

data (数据元素)      next (游标)

王道考研/CSKAOYAN.COM

4



### 用代码定义一个静态链表

内存

```

#define MaxSize 10
typedef struct {
    ElemType data;
    int next;
} SLinkList[MaxSize];
        
```

等价

```

#define MaxSize 10
struct Node{
    ElemType data;
    int next;
};
typedef struct Node SLinkList[MaxSize];
        
```

```

void testSLinkList() {
    SLinkList a;
    //.....后续代码
}
        
```

等价

```

void testSLinkList() {
    struct Node a[MaxSize];
    //.....后续代码
}
        
```

a 是一个静态链表

a 是一个 Node 型数组

可用 SLinkList 定义“一个长度为 MaxSize 的 Node 型数组”

王道考研/CSKAOYAN.COM

5

### 对猜想的验证

```

#define MaxSize 10
struct Node{
    int data;
    int next;
};
typedef struct {
    int data;
    int next;
} SLinkList[MaxSize];

void testSLinkList() {
    struct Node x;
    printf("sizeX=%d\n", sizeof(x));

    struct Node a[MaxSize];
    printf("sizeA=%d\n", sizeof(a));

    SLinkList b;
    printf("sizeB=%d\n", sizeof(b));
}
        
```

结论:

SLinkList b —— 相当于定义了一个长度为 MaxSize 的 Node 型数组

运行结果

```

sizeX=8
sizeA=80
sizeB=80
Process finished with exit code 0
        
```

王道考研/CSKAOYAN.COM

6



静态链表

a

0	头	-1
1		
2		
3		
4		
5		
6		
7		
8		
9		

数据元素

内存

```
#define MaxSize 10
typedef struct {
    ElemType data;
    int next;
} SLinkList[MaxSize];
```

初始化静态链表:  
把 a[0] 的 next 设为 -1  
把其他结点的 next 设为一个特殊值用来表示结点空闲, 如 -2

下一个节点的数组下标 (游标)

```
//静态链表的最大长度
//静态链表结构类型的定义
//存储数据元素
//下一个元素的数组下标
void testSLinkList() {
    SLinkList a;
    //.....后续代码
}
```

L

头

NULL

单链表

王道考研/CSKAOYAN.COM

7

静态链表

a

0	头	2
1	e <sub>2</sub>	6
2	e <sub>1</sub>	1
3	e <sub>4</sub>	-1
4		
5		
6	e <sub>3</sub>	3
7		
8		
9		

数据元素

内存

```
#define MaxSize 10
typedef struct {
    ElemType data;
    int next;
} SLinkList[MaxSize];
```

查找:  
从头结点出发挨个往后遍历结点

插入位序为 i 的结点:  
①找到一个空的结点, 存入数据元素  
②从头结点出发找到位序为 i-1 的结点  
③修改新结点的 next  
④修改 i-1 号结点的 next

如何判断结点是否为空?  
①从空结点中找下一个非空结点  
②从非空结点中找下一个空结点

删除某个结点:  
①从非空结点中找下一个非空结点  
②修改前驱结点的游标  
③被删除结点 next 设为 -2

```
//静态链表的最大长度
//静态链表结构类型的定义
//存储数据元素
//下一个元素的数组下标
void testSLinkList() {
    SLinkList a;
    //.....后续代码
}
```

①从非空结点中找下一个非空结点  
②修改前驱结点的游标  
③被删除结点 next 设为 -2

王道考研/CSKAOYAN.COM

8



### 知识回顾与重要考点

#### 静态链表

下标	数据域	指针域
0	头	2
1	e <sub>2</sub>	6
2	e <sub>1</sub>	1
3	e <sub>4</sub>	-1
4		-2
5		-2
6	e <sub>3</sub>	3
7		-2
8		-2
9		-2

0号结点充当“头结点”

游标为-1表示已经到达表尾

游标充当“指针”

用一个特殊的数值标记空闲结点

下一个结点的数组下标(游标)

```

#define MaxSize 10
typedef struct {
    ElemType data;
    int next;
} SLinkList[MaxSize];

void testSLinkList() {
    SLinkList a;
    //.....后续代码
}

```

//静态链表的最大长度

//静态链表结构类型的定义

//存储数据元素

//下一个元素的数组下标

静态链表：用数组的方式实现的链表

优点：增、删操作不需要大量移动元素

缺点：不能随机存取，只能从头结点开始依次往后查找；容量固定不可变

适用场景：①不支持指针的低级语言；②数据元素数量固定不变的场景（如操作系统的文件分配表FAT）

王道考研/CSKAOYAN.COM

9

@王道论坛

@王道计算机考研备考

@王道咸鱼老师-计算机考研

@王道楼楼老师-计算机考研

@王道计算机考研

@王道计算机考研

@王道计算机考研

@王道在线

10